

# Quarch Technology Ltd

# Torridon Communications

# Manual

For use with the Torridon System

Tuesday, 18 March 2014



## Change History

1.0	14th March 2012	First Version
1.2	23 <sup>rd</sup> April 2012	Added Python Examples
1.3	17 March 2014	Minor corrections and updates

## Contents

Introduction to the Torridon System .....	5
Connection Tables .....	6
Direct Controlled Modules .....	7
Direct Control Example .....	8
Array Controlled Modules .....	8
Array Connection Examples .....	9
Terminal Types .....	11
Serial Control .....	11
Connection Parameters.....	11
HyperTerminal Example.....	12
Putty Terminal Example .....	14
TerraTerm Terminal Example.....	15
USB Control.....	16
Installing USB Drivers .....	16
Telnet Control.....	18
Connection Parameters.....	18
Device Types .....	19
Modules.....	19
Controllers: .....	20
Interface Kit (QTL1260) .....	20
Array Controller (QTL1079, QTL1461).....	20
TestMonkey .....	22
Getting Started .....	23

Torridon Terminal .....	25
Torridon API .....	26
PERL Scripting.....	27
Installing Perl on Windows .....	27
Python Scripting.....	28
Installing Python on Windows .....	28

## Introduction to the Torridon System

The Torridon System is a modular based test system with a number of parts that can be used in many different configurations.

In all cases, the Torridon System is controlled by ASCII terminal commands. These are simple, readable, text commands that can be sent to a device. Responses from a command are in the same simple text format.

Each device supports one or more of Serial, USB or Ethernet (Telnet) control.

- 'Lite' Modules support only Serial.
- Standard Modules support Serial and USB.
- Array Controllers, and some advanced Modules, also support Telnet.

Any terminal command must be 64 characters or less in length. Commands can return any number of lines of data in response to a command, but each line will also be limited to 64 characters in length.

## Connection Tables

S = Serial U = USB T = Telnet	QTL1260 - Interface Kit					QTL1462 - 4 Port Array Controller				QTL1079 - 28 Port Array Controller			
	Serial	USB-Serial	USB	Telnet	TestMonkey 2	Serial	USB	Telnet	TestMonkey 2	Serial	USB	Telnet	TestMonkey 2
Lite	✓	✓	✗	✗	S/U	✓	✓ <sup>1</sup>	✓	S/U/T	✓	✓ <sup>1</sup>	✓	S/U/T
Standard	✓	✓	✓ <sup>1</sup>	✗	S/U	✓	✓ <sup>1</sup>	✓	S/U/T	✓	✓ <sup>1</sup>	✓	S/U/T
HS	✓	✓	✓ <sup>1</sup>	✗	S/U	✓	✓ <sup>1</sup>	✓	S/U/T	✓	✓ <sup>1</sup>	✓	S/U/T
SBB	✓	✓	✓ <sup>1</sup>	✗	S/U	✓	✓ <sup>1</sup>	✓	S/U/T	✓	✓ <sup>1</sup>	✓	S/U/T
Cable Pull	✓	✓	✓ <sup>1</sup>	✗	S/U	✓	✓ <sup>1</sup>	✓	S/U/T	✓	✓ <sup>1</sup>	✓	S/U/T
MUX	✓	✓	✓ <sup>1</sup>	✗	S/U	✓	✓ <sup>1</sup>	✓	S/U/T	✓	✓ <sup>1</sup>	✓	S/U/T

NOTES:

1. USB connections can only be used with Quarch software (TestMonkey 2, Torridon Terminal) or with applications built with the Torridon API.
2. A small number of stand-alone modules are available; these have a variety of different interface options. See their data sheet for details

## Direct Controlled Modules

A module can be controlled by sending commands to it. This requires a direct Serial/USB/Telnet connection to the module. This could be obtained by:

- Connecting a module to your PC, using an Interface Kit.
- Using a 'stand-alone' Module that has a Serial/USB/Ethernet(Telnet) port on it.

This is a simple and fast way to control a single module. It may be complex or impractical to control a large number of modules in this way however.

## Direct Control Example

A QTL1253 (Mini SAS Cable Pull Module) is connected to a windows PC, via its **Serial** port.

A Terminal connection is opened using **HyperTerminal** and user types the command “**hello?**” at the cursor. This is a standard command that will respond with the name of the module.

The response is:

```
>hello?  
mini SAS Cable Pull Module  
>
```

We can see that the module responded directly to the command that was sent. All other commands that the module supports can be sent in the same way.

## Array Controlled Modules

When you want to control several modules at the same time, we provide Array Controllers. These allow multiple modules to be controlled as a single test system. They are described in more detail later in this document.

The module(s) are no longer connected to your computer. Instead they are connected to the Array Controller. The Controller is then connected to your computer via USB/Serial/Telnet.

## Array Connection Examples

A QTL1253 (Mini SAS Cable Pull Module) is connected to **Port 1** of an Array Controller. A QTL1177 (SAS/SATA Drive Control Module) is connected to **Port 2** of the same Array Controller.

As before, the Controller is connected to a windows PC, via its **Serial** port. A Terminal connection is opened using **HyperTerminal** and user types the command "**hello?**" at the cursor.

The response is:

```
>hello?  
4 Port Torridon Array Controller  
>
```

This time, the Module did NOT respond. We are not talking to either of the attached modules, but to the Array Controller.

To send a command to a module, we need to 'address' the one(s) we want to talk to. This is done by adding an 'Address List' to the end of the command:

```
>hello? <1>  
1: mini SAS Cable Pull Module  
>hello? <2>  
2: SAS/SATA Drive Control Module
```

Here we have sent the same "hello?" command to each of the attached modules in turn. The address number is enclosed in angle brackets. The response has also changed. Now we have "**1:**" appended to the response for the Module on **Port 1**. This confirms that the response came from that module.

Multiple modules can be addressed at the same time. This can be used to speed up testing or to ensure that actions happen at the same time:

```
>hello? <1,2>  
1: mini SAS Cable Pull Module  
2: SAS/SATA Drive Control Module  
>
```

This command addresses both the Module on **Port 1** and **Port 2**. This is called using an 'Address List'. Now you can see that the number at the start of the response line is critical to tell which response comes from which port.

Address Lists can be complex, including lists <1,2>, ranges <1-3> or both <1-3,4,5-7>. The entire length of the command and the address list must not exceed 64 characters.

## Terminal Types

As described in the introduction, there are options for Serial, USB and Telnet control. Here we will look at each of the options and how to use it.

### Serial Control

A serial terminal is the simplest method of control. This can be from a Serial port on a Windows/Linux/Mac computer or from an external Serial control system such as a serial Terminal Server. Most Torridon modules support a serial Terminal.

#### Connection Parameters

All Torridon devices use the same serial connection parameters:

Baud Rate	<b>19,200</b>
Data Bits	<b>8</b>
Parity	<b>NONE</b>
Stop Bits	<b>1</b>
Flow Control	<b>NONE</b>

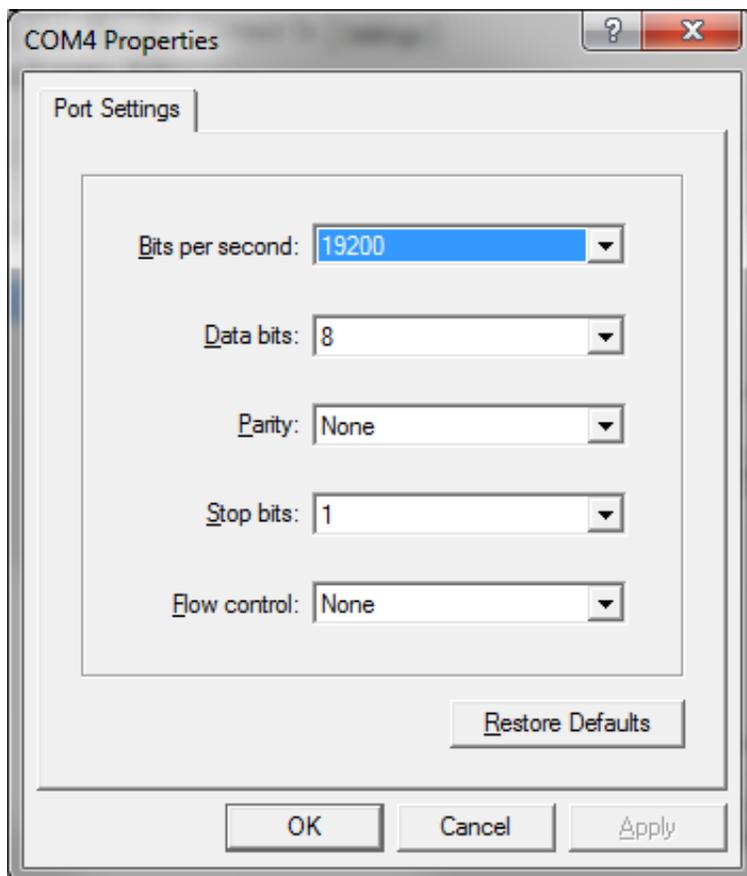
There are a few Terminal settings that you can alter if desired, see the Technical Manual for your device:

- Array Controllers DO support hardware flow control on the Serial connection, but this is not the default and must be enabled through a separate command.
- By default, the terminal will echo back all typed characters, the terminal echo and other settings can be altered if required.

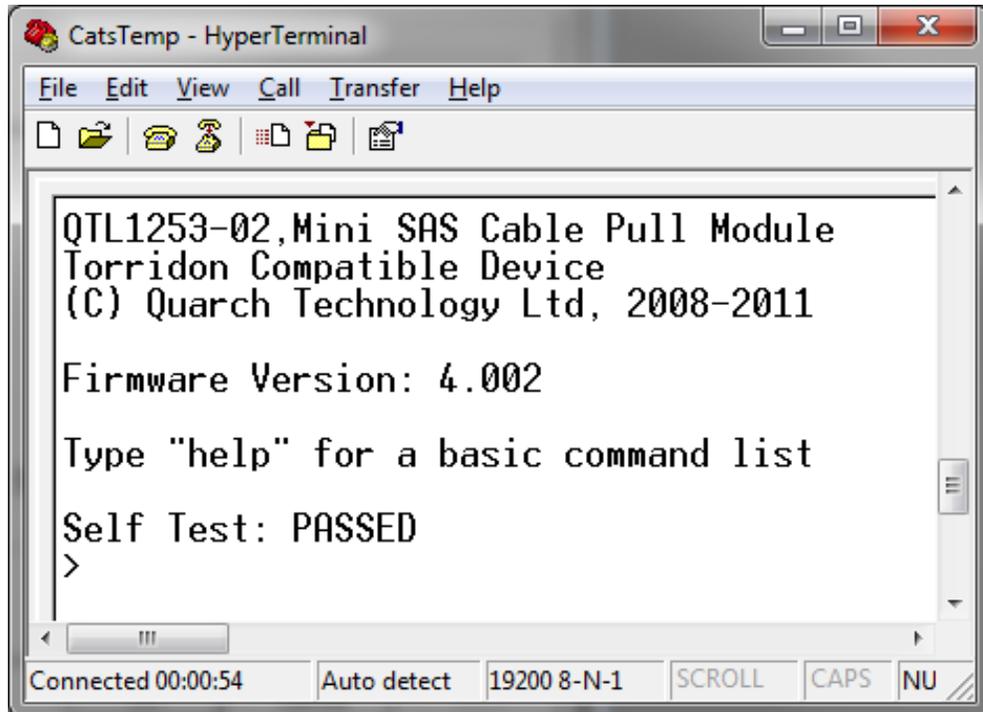
## HyperTerminal Example

Many Windows PCs come with HyperTerminal installed; this is a simple application that can connect to a Serial Terminal.

- Open HyperTerminal
- Select **File->New Connection** and enter a name for the connection
- Select the **COM** port that your device is connected to and click **OK**
- Set the Port Settings as shown below:



- Press 'Enter'. If the module is correctly connected and powered up, you should see the start screen as shown below:



```
QTL1253-02, Mini SAS Cable Pull Module
Torridon Compatible Device
(C) Quarch Technology Ltd, 2008-2011

Firmware Version: 4.002

Type "help" for a basic command list

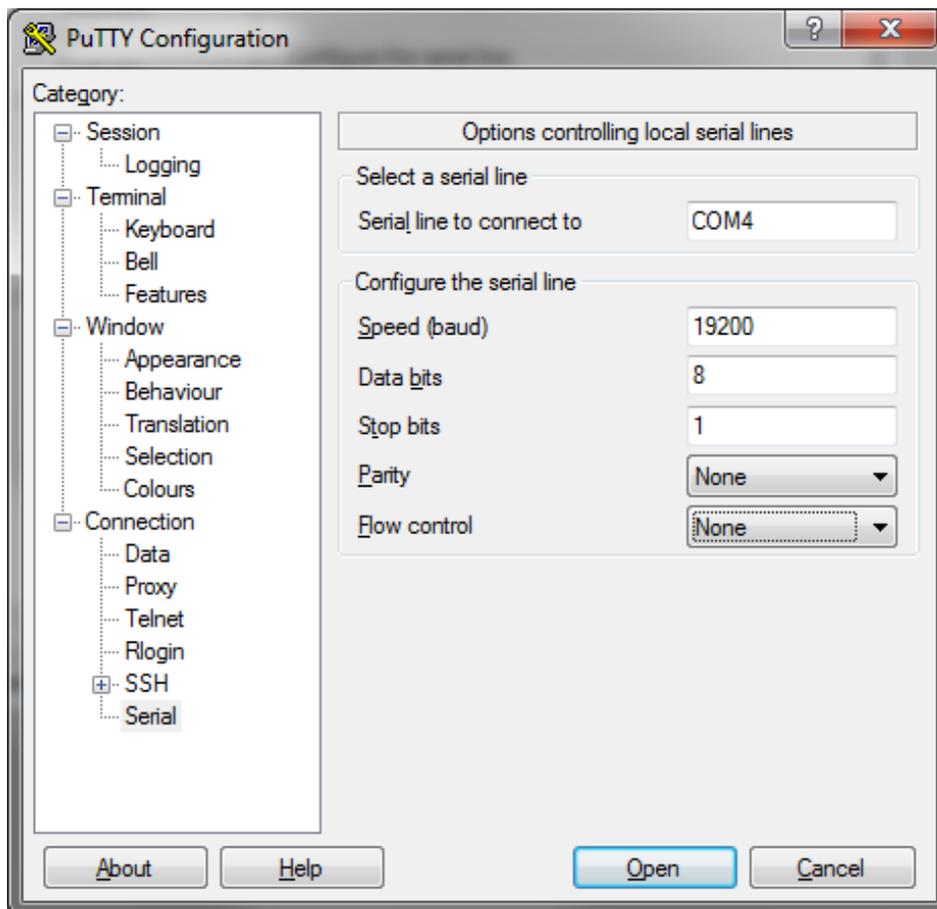
Self Test: PASSED
>
```

- You can now begin sending commands to control the device

## Putty Terminal Example

PuTTY is a free terminal application that is available for several operating systems.

- Open Putty and select **Session** from the **Category** selection.
- Select **Connection Type->Serial**
- Select **Serial** on the Category selection
- Set the options as shown below:

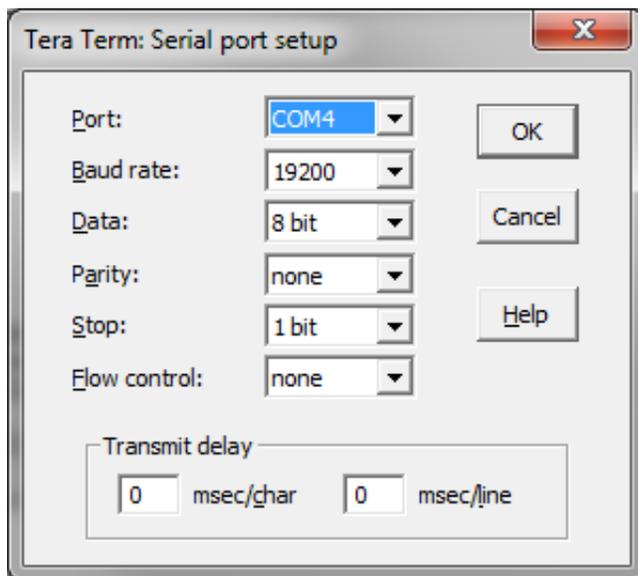


- Click **Open** and the Terminal window should be shown
- Press 'Enter'. If the module is correctly connected and powered up, you should see the start screen
- You can now begin sending commands to control the device

## TerraTerm Terminal Example

TerraTerm is a free terminal application that is available for several operating systems.

- Open TerraTerm and select **File->New Connection**
- Select **Serial** then select the **COM port**
- Select **Setup->SerialPort...**
- Set the options as shown below:



- Click **OK** and the Terminal window should be shown
- Press 'Enter'. If the module is correctly connected and powered up, you should see the start screen
- You can now begin sending commands to control the device

## USB Control

USB control is available on many Torridon Modules. To connect to a Module using USB, you will require software from Quarch. Unlike Serial and Telnet, there is no 'standard' Terminal application available. Your current choices are:

- **TestMonkey** : This is our main GUI control application, it is designed to directly connect to a single module and allow easy manual control. TestMonkey is described in detail later.
- **Torridon Terminal** : This is our simple Terminal application. It allows you to connect to a Torridon device through USB (as well as Serial and Telnet). Commands are typed by the user in the same way as with HyperTerminal or similar. Torridon Terminal is described in detail later.
- **Torridon API** : A C# based API allows USB (as well as Serial and Telnet) control of a module. The API is free and can be used to create your own applications. Please contact [support@quarch.com](mailto:support@quarch.com) for details on the API.

## Installing USB Drivers

USB control of a device will require the USB driver to be installed first. Torridon devices use a standard WinUSB driver provided by Microsoft.

Download the latest driver from: <http://quarch.com/downloads.html>

Run the correct installer for the version of Windows you are using (there are 3 options). If you try the wrong version, you will get an error message and can choose a different version.

You will be warned that the driver is not signed, ignore this and continue.

The driver should be installed without any Torridon devices attached to your PC. If you have devices attached, you will have to re-plug them after installing.

### *Installing on Windows XP*

Due to our driver not being signed, it will not install on XP by the above method. Instead follow these steps:

1. Attach the device to the PC and power it up
  - a. Windows should pop-up **"New Hardware Found"**
  - b. The **"Found new hardware"** wizard will start
2. Select **"Install from a specified location"**
3. Browse to the folder containing the driver files and select it
4. You will be warned that the driver **"Has not passed windows logo testing"**
  - a. Select **"Continue Anyway"**
5. The driver should not install with no further prompts, click **Finish** when done
6. Open device manager and check the device is visible without a warning icon
  - a. Open the device and go to the **Drivers** tab.
  - b. Check the driver version is 1.0.2.0
  - c. Select **Driver Details** and check the provider is **Microsoft** and that **WisUSB.SYS** is in the file list.
7. The install was successful and the device should now work correctly

## Telnet Control

Torridon Array Controllers and a small number of specialist modules support a Telnet Terminal. As with Serial, you can use a number of standard Terminal clients to connect to the module.

### Connection Parameters

Connection Type	<b>TCP/IP</b> (SSL is not supported)
Port	<b>23</b>
Target	<b>IP Address</b> or <b>NetBios</b> Name

- Torridon devices support **DHCP** and will be assigned an IP address by your network if possible. If this fails, the default IP is **192.168.1.99**.
- Torridon devices support **NetBios** names. If your network allows this, you can connect to the device with its name. By default this is its serial number in the form "**QTLnnnnn-nn-*nnn***". See the label attached to the device.
- If you are connected to the device by other means you can get the current IP address with the command "**CONFig:ETHernet IP?**". The current NetBios name can be returned with "**CONFig:ETHernet NAME?**".

Currently our Telnet server does not support the entire Telnet auto-negotiation protocol. As such, you may have to override some settings in your Telnet client to ensure the connection works fully.

- Turn off local echo of typed characters
  - o If this is not possible, the command "**CONFig:TERMinal SCRIPT**" will alter the terminal mode on the Array to prevent it from echoing characters.
- Transmitted lines should be terminated by CR+LF
- If you will be sending commands very rapidly, you may need to set a 100mS delay between each line sent out to allow it to avoid overloading the Telnet server

The Telnet server has been tested with HyperTerminal, Putty, TerraTerm, Torrison Terminal and Microsoft Telnet.exe

For manual terminal control, we recommend using Torrison Terminal or Putty (if you require a cross-platform application).

## Device Types

The Torridon system is made up of **Modules** which perform the tests/actions, and **Controllers** which provide power and control to the modules.

### Modules

A wide range of test modules are available, from hot-swap drive modules, cable-pull modules, Physical Layer Switch (Mux) modules and more.

- Most modules have are connected via a thin Torridon flex cable that provides both power and control. These require a 'Controller' (see below)
- Some modules are 'Stand-alone'. These have some combination of USB/Serial/Ethernet ports for control and are power from a separate power supply. These modules do NOT require a controller, though some can be used with one (if they have a Torridon flex cable output)

## Controllers:

Controllers provide power and/or control for modules. These include:

### Interface Kit (QTL1260)

This controller provides power and control to a single Torridon Module. It comes with a 12v power supply and cables for Serial (DB-9) and USB. This Kit is normally used when you want to control one module for manual testing or as part of a small automated test setup. **TestMonkey** is often used for manual testing and **PERL** scripts or similar for automated testing

Communication methods are:

- **USB:** This requires a module that supports USB (most modules except 'Lite' modules). To use this mode, connect a USB capable module to the interface kit and a USB cable to the PC. The button on the interface kit must be **OUT**. The 12v power supply must be connected to the interface kit.
- **USB-Serial:** In this mode, a virtual COM port appears on your PC, allowing you to communicate to the module on serial even if your PC does not have a serial port. This mode is supported by all standard modules. To use this mode, connect any module to the interface kit and a USB cable to the PC. The button in the interface kit must be **IN**. The USB drivers will install automatically if you have an internet connection. The 12v power supply must be connected to the interface kit.

Note: If you need to download the drivers manually, they can be found at:

<http://www.ftdichip.com/Drivers/VCP.htm>

- **Serial:** If your PC has a Serial port, you can use the included cable to control the module. A Terminal Server can also control the module via serial though you may require us to provide a different pin-out for the serial cable. Connect any module to the interface kit. The 12v power supply must also be connected to the interface kit. Connect the orange serial cable to the RS-232 connection on the interface kit. Connect the other end of the cable to your PC serial port. The button on the interface kit must be **OUT**.

### Array Controller (QTL1079, QTL1461)

Array Controllers allow several modules to be controlled at the same time, from a single place. They also greatly reduce cable complexity by removing the need for separate power supplies and control cables. This is normally used for large/complex test systems which are controlled by automated test scripts.

The controller uses a single 12v power supply to power all the modules. The QTL1079 (28-Port Controller) is designed to run drive modules. While it can run any combination of modules, it is limited to 60 watts in total which may limit the number of other modules you can connect at a time.

- **Serial:** You can use the included cable to control the Array and all its attached modules from a single serial port.
- **USB:** The Array Controllers have internal USB hubs, this will allow any attached module (that supports USB control) to appear on your PC as if it was directly connected with an Interface Kit. In this case the section '**Direct Controlled Modules**' above describes how the modules are controlled.

In addition, newer Array Controllers (Not the QTL1079), can support their own USB Terminal connection. In this case the section '**Array Controlled Modules**' above describes how the modules are controlled.

Note that USB is unique in this regard. Both Serial and Telnet connections are made to the Array Controller and NOT directly to the modules. USB is different due to the internal USB hubs within the Array Controllers. These hubs are required to allow Firmware update of Modules by USB but also provide another communications option.

- **Telnet:** You can attach the Array Controllers to your internal LAN to control them easily from a distance.

## TestMonkey

TestMonkey is a GUI application for Windows that provides a quick way to get started with Torridon modules. It is designed to control a single Torridon Module at a time.

- A later release this year will allow control of multiple modules
- Currently all modules must be 'directly connected' to the PC (for example a module attached to an Array Controller on Telnet cannot be controlled). This will be addressed in a release later this year.

Currently, TestMonkey can control:

- A standard module attached with an Interface Kit (on Serial or USB)
- A Lite module attached with an Interface Kit (on Serial)
- A Stand-alone module attached directly (on Serial USB or Telnet)
- A standard module attached to an Array Controller (via internal USB hubs)

Currently, TestMonkey can NOT control:

- 2 modules at the same time
- A Lite module connected to an Array Controller
- Any module on an Array Controller, if the Controller is attached to the PC by Serial or Telnet

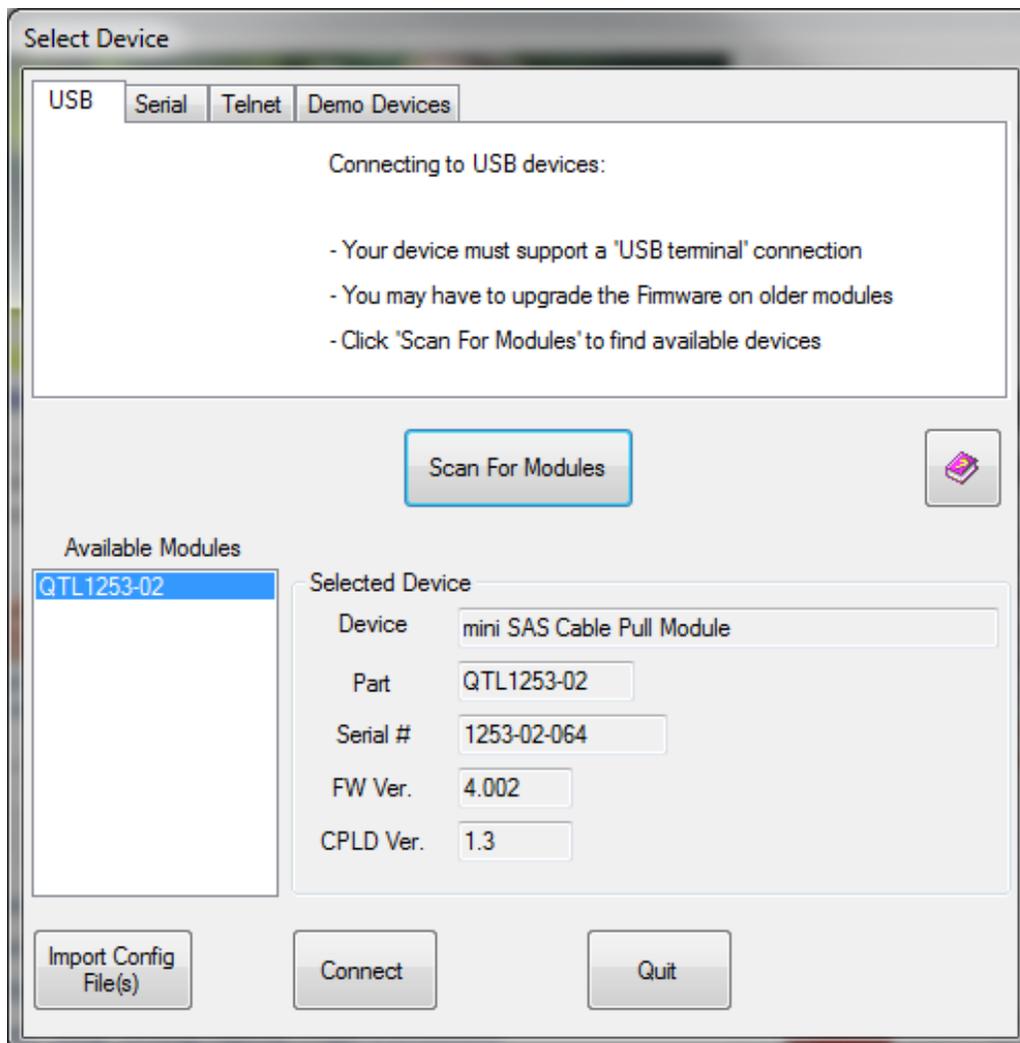
This assumes that the modules are using the latest available Firmware version. Some old Firmware versions may not work fully with TestMonkey. You can contact [support@quarch.com](mailto:support@quarch.com) for assistance in updating your Firmware.

## Getting Started

Download the latest version of TestMonkey from: <http://quarch.com/downloads.html>

Install and run the application. If you intend to use USB control, you will first need to install the USB drivers as described above.

When you run TestMonkey, you will be prompted to select a module to use. First select the connection method (USB/Telnet/Serial) that you require then select '**Scan for Modules**':



If you have more than one module attached, you will have to select it from the list of '**Available Modules**'. You can identify the correct module by its description and serial number, shown to the right.

For the TestMonkey manual, click the manual icon on the connection screen or select **Help->Help** from the main menu of the application. You can also download a paper copy of the TestMonkey 2 manual from: <http://quarch.com/downloads.html> in the '**Torridon Manuals**' section.

## Torridon Terminal

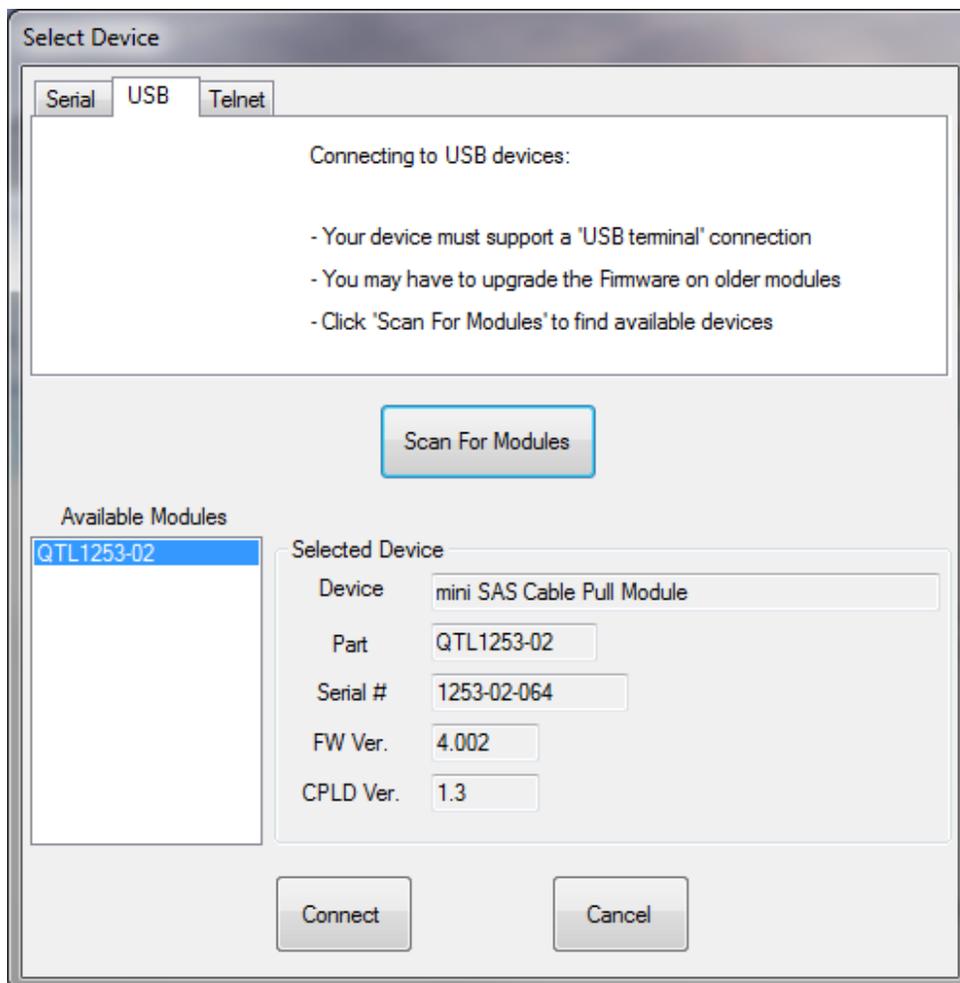
Torridon Terminal is simple terminal application, similar to HyperTerminal. It runs on Windows and requires the .NET 2.0 Framework to be installed.

Torridon Terminal makes it easier to identify the module you want to use. It can also control a module that is connected on USB. This is not something that can be done with a traditional terminal client.

Download it from <http://quarch.com/downloads.html> in the '**Torridon Applications**' section.

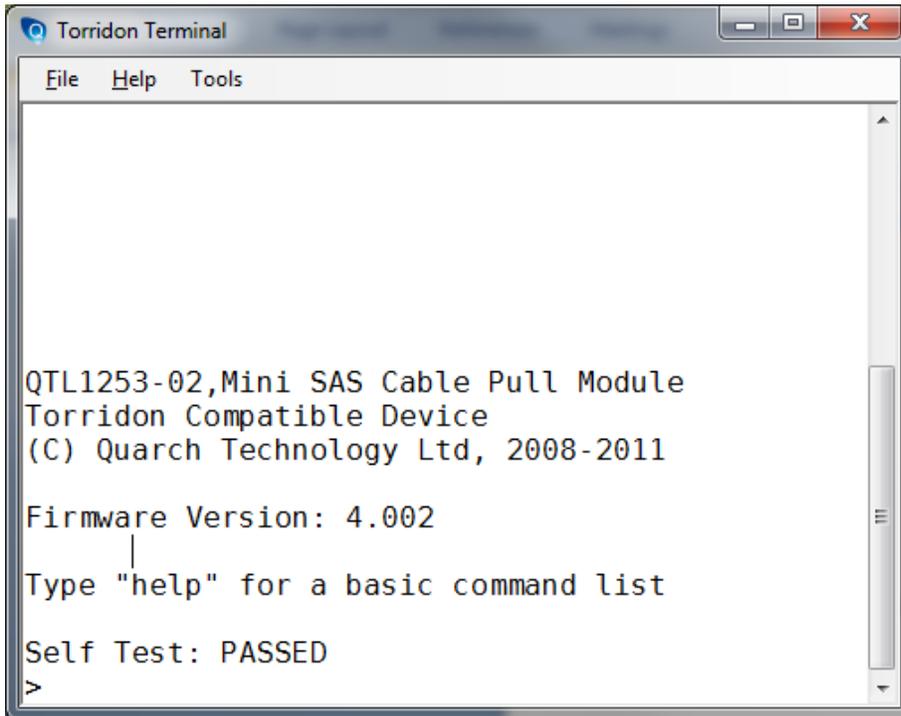
Install and run the application. If you intend to use USB control, you will first need to install the USB drivers as described above.

When you run TestMonkey, you will be prompted to select a module to use. First select the connection method (USB/Telnet/Serial) that you require then select '**Scan for Modules**':



If you have more than one module attached, you will have to select it from the list of '**Available Modules**'. You can identify the correct module by its description and serial number, shown to the right.

When you select **Connect**, the Terminal window will be shown. Type commands as normal.



```
Torrison Terminal
File Help Tools

QTL1253-02,Mini SAS Cable Pull Module
Torrison Compatible Device
(C) Quarch Technology Ltd, 2008-2011

Firmware Version: 4.002
Type "help" for a basic command list
Self Test: PASSED
>
```

## Torrison API

The Torrison API is written in C# and is the basis for all control used in TestMonkey 2 and Torrison Terminal.

The API is a VS2005 project and built against NET 2.0

Please contact [support@quarch.com](mailto:support@quarch.com) for a copy of the API and associated examples.

## PERL Scripting

Perl scripts are a simple and common way to control Torridon Modules. If your current test automation system uses Perl already, this will probably be the best way to work.

Perl cannot control Torridon devices over USB, you will need to use a Serial or Telnet connection.

### Installing Perl on Windows

We use the 32 bit version(x86) of ActiveState Perl for development testing. Download it from:

<http://www.activestate.com/activeperl/downloads>

You will have to re-boot pc after installing perl

To use the serial port, you will need to install the **win32::SerialPort** module for Perl. This is not always found in the Perl Package Manager. You may need to install it manually:

- Download from: <http://search.cpan.org/~bbirth/Win32-SerialPort-0.22/lib/Win32/SerialPort.pm>
- Follow the instructions in the included text file.

On Linux systems you will need the Device::SerialPort module. The calls are identical for the Windows and Linux modules so you will only need to change the '**use**' instruction at the top of the files to use our examples in Linux.

We supply a Perl module that makes it easy to send commands to the modules:

<b>TorridonCommon.pl</b>	Common functions for Serial communications
<b>TorridonCommon_Telnet.pl</b>	Common functions for Telnet communications

Chose the module for the type of connection you require. We also have a number of sample scripts that show the files in use.

Download them from <http://quarch.com/downloads.html> in the '**Torridon Applications**' section.

## Python Scripting

Python scripts are another simple and common way to control Torridon Modules.

Python cannot control Torridon devices over USB, you will need to use a Serial or Telnet connection.

### Installing Python on Windows

We use the 32 bit version of ActiveState Python 3.2 or higher for development testing. Download it from:

<http://www.activestate.com/activepython/downloads>

To use the serial port you will also need the pyserial module

1. Run the Python Package Manager (PyPm)
2. Run the command “pypm install pyserial”

Active Python comes with the Win32 extensions. If you used a different version of Python, you may need to install the extensions manually from:

<http://sourceforge.net/projects/pywin32/files/pywin32/>

We supply Python examples that makes it easy to send commands to the modules:

<b>BasicSerialComms.py</b>	Serial port example
<b>BasicTelnetComms.py</b>	Telnet example

Download them from <http://quarch.com/downloads.html> in the '*Torridon Applications*' section.